Rapid Algorithm for Computing the Electron Repulsion Integral over Higher Order Gaussian-Type Orbitals: Accompanying Coordinate Expansion Method

KAZUHIRO ISHIDA

Department of Chemistry, Faculty of Science, Science University of Tokyo, Shinjuku-ku, Tokyo 162, Japan

Received 3 July 1997; accepted 16 January 1998

ABSTRACT: A general algorithm for rapidly computing the electron repulsion integral (ERI) is derived for the ACE-b3k3 formula, which has been derived previously. [K. Ishida, *Int. J. Quantum Chem.*, **59,** 209 (1996)]. A computer program code that is universal for all types of Gaussian-type orbitals (GTOs) up to *h*-type can be constructed by the use of this general algorithm. It is confirmed that the ACE-b3k3 algorithm is numerically very stable even for higher order GTOs. It is found that, in a floating-point-operation (FLOP) count assessment, the ACE-b3k3 algorithm is the fastest among all methods available in the literature for $(dd|dd)$, $(ff|ff)$, $(gg|gg)$, and $(hh|hh)$ ERIs when the degree of contraction of the GTO is high. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 923–934, 1998

Keywords: molecular integral; electron repulsion integral; rapid algorithm; accompanying coordinate expansion method

Introduction

The rapid and rigorous calculation of the electron repulsion integral (ERI) is desired in both SCF and post-SCF calculations. Recently, several investigators[1] have tried to evaluate ERI approximately for its rapid computation. However, these approximations are suitable only for crude calculations, because the precision of the approximation is not sufficient.[1] For example, in the resolution of the identity (RI) approximation,[1] the product of Gaussian-type orbitals (GTOs) is expanded in terms of basis set. Thus, the error of the RI approximation is similar to the basis-set truncation error. As a result, a high-quality basis set, like

Correspondence to: K. Ishida

CCC 0192-8651 / 98 / 080923-12

the near-Hartree–Fock method, is necessary to change the RI approximation into a reliable one.

In a previous study,[2] we derived the accompanying coordinate expansion (ACE) formula, which is a series of general ERI formulas. These ACE formulas can calculate ERI in any desired precision, because these are rigorous general formulas. The ACE algorithm obtained from ACE formulas is the fastest for computing ERI of *s*- and *p*-type GTOs in the FLOP count assessment, as shown in a previous study.[2] Among ACE formulas, the ACE-b3k3 formula is used for the case when the degree of contraction of GTOs, say K, is larger. For the case when K is smaller, it is proper to use the other ACE formulas; for example, b2k3, b1k1, and so on.[2] A general ACE-b3k3 algorithm was derived from the expanding ACE-b3k3 general formula for the individual cases as $(pp|pp)$ and $(dd|dd)$ classes of ERIs. However, this expansion needs a large amount of tedious manipulation; therefore, it was not done for GTOs of higher order than *f*-type.

In this article, extension is done for $(ff|ff)$, $(gg|gg)$, and $(hh|hh)$ ERIs, with the result that the previous general algorithm must be revised for extension of GTOs of up to *h*-type. The old algorithm is still valid for GTOs of up to *d*-type. The new algorithm is then shown. A computer program code, which is universal for all types of GTOs up to *h*-type, is made by the use of this new algorithm. Numerical examples for the use of this program code are shown for $(LL|LL)$ ERIs (L = 2–5) in the final section.

ACE-b3k3 General Algorithm

When the Cartesian GTOs in an ERI are centered at **A**, **B**, **C**, and **D** and have the exponents α_A, α_B, α_C, and α_D and the so-called quantum numbers $(l_A m_A n_A)$ and so on, the ACE-b3k3 general formula of the ERI can be written as[2]:

$$\mathrm{ERI} = A_0 \sum_{\mathbf{N}_3} C_4\{\mathbf{N}_3\} H_4\{\mathbf{N}_3\} \tag{1}$$

where:

$$A_0 = \left(2\pi^{5/2}/\gamma_1\gamma_2\sqrt{\gamma_1+\gamma_2}\right) \times \exp\left(-\alpha_A\alpha_B\overline{AB}^2/\gamma_1 - \alpha_C\alpha_D\overline{CD}^2/\gamma_2\right) \tag{2}$$

in which $\gamma_1 = \alpha_A + \alpha_B$ and $\gamma_2 = \alpha_C + \alpha_D$. The accompanying coordinate part is given by:

$$C_4\{\mathbf{N}_3\} = \sum_{\{\mathbf{M}\}} D^{A3}_{i'i''} D^{B3}_{j'j''} D^{C3}_{k'k''} D^{D3}_{h'h''} \tag{3}$$

where $\{\mathbf{M}\} = \{\mathbf{M}_A \mathbf{M}_B \mathbf{M}_C \mathbf{M}_D\}$, $\mathbf{M}_A = (M_{Ax} M_{Ay} M_{Az})$, $0 \le M_{Ax} \le l_A$, $0 \le M_{Ay} \le m_A$, and $0 \le M_{Az} \le n_A$:

$$D^{A3}_{i'i''} = \sum_{\{\mathbf{i}'\}} \sum_{\{\mathbf{i}''\}} D^{A3x}_{i'_x i''_x} D^{A3y}_{i'_y i''_y} D^{A3z}_{i'_z i''_z} \tag{4}$$

and:

$$D^{A3x}_{i'_x i''_x} = \binom{l_A}{M_{Ax}} \binom{l_A - M_{Ax}}{i'_x} \times \binom{i'_x}{i''_x} AB_x^{l_A - M_{Ax} - i'_x} CD_x^{i'_x - i''_x} AC_x^{i''_x} \tag{5}$$

in which: $\{\mathbf{i}''\} = \{i''_x i''_y i''_z\}$, $\{\mathbf{i}'\} = \{i'_x i'_y i'_z\}$, $0 \le i'_x \le l_A - M_{Ax}$, and $0 \le i''_x \le i'_x$. The assembly of indices $\{\mathbf{N}_3\}$ is $\{M_A M_B M_C M_D i'j'k'h'i''j''k''h''\}$, where $0 \le M_A \le L_A$, $0 \le M_B \le L_B$, $0 \le M_C \le L_C$, $0 \le M_D \le L_D$, $0 \le i' \le L_A - M_A$, $0 \le j' \le L_B - M_B$, $0 \le k' \le L_C - M_C$, $0 \le h' \le L_D - M_D$, $0 \le i'' \le i'$, $0 \le j'' \le j'$, $0 \le k'' \le k'$, $0 \le h'' \le h'$, $M_A = M_{Ax} + M_{Ay} + M_{Az}$, $i' = i'_x + i'_y + i'_z$, $i'' = i''_x + i''_y + i''_z$, and so on. The $(L_A L_B L_C L_D)$ is the angular momentum quartet of the ERI requested.

The core part is given by:

$$\begin{aligned} H_4\{\mathbf{N}_3\} &= \sum_{i_1}\sum_{i_2} G_{i_1 i_2} \sigma_B^{a_B + c_B} \sigma_A^{b_A + d_A} \sigma_D^{c_D + a_D} \sigma_C^{d_C + b_C} \\ &\times \sigma_1^{M_A + M_B - i_1} \sigma_2^{M_C + M_D - i_2} / (\sigma_1 + \sigma_2)^{M - i_1 - i_2} \\ &\times \sum_{s_1}\sum_{s_2} (-)^{s_1 + s_2} \binom{a_B + b_A}{s_1} \binom{c_D + d_C}{s_2} \\ &\times \left(\frac{\sigma_1}{\sigma_1 + \sigma_2}\right)^{i' + j'} \left(\frac{\sigma_2}{\sigma_1 + \sigma_2}\right)^{k' + h'} F_{\nu + s_1 + s_2}(z) \end{aligned} \tag{6}$$

where $\sigma_B = \alpha_B/\gamma_1$, $\sigma_A = \alpha_A/\gamma_1$, $\sigma_D = \alpha_D/\gamma_2$, $\sigma_C = \alpha_C/\gamma_2$, $a_B = L_A - M_A - i'$, $b_A = L_B - M_B - j'$, $c_D = L_C - M_C - k'$, $d_C = L_D - M_D - h'$, $a_D = i' - i''$, $b_C = j' - j''$, $c_B = k' - k''$, $d_A = h' - h''$, $\sigma_1 = 1/2\gamma_1$, $\sigma_2 = 1/2\gamma_2$, $0 \le i_1 \le \mathrm{int}[(M_A + M_B)/2]$, $0 \le i_2 \le \mathrm{int}[(M_C + M_D)/2]$, $M = (M_A + M_B + M_C + M_D)/2$, $0 \le s_1 \le a_B + b_A$, $0 \le s_2 \le c_D + d_C$,

$\nu = i' + j' + k' + h' + M - i_1 - i_2$, and:

$$G_{i_1 i_2} = (-)^{M_A + M_B + M - (i_1 + i_2)} \sum_{\{\mathbf{I}\}} g_x g_y g_z \quad (7)$$

in which $\{\mathbf{I}\} = \{i_{1x} i_{1y} i_{1z} i_{2x} i_{2y} i_{2z}\}$, $0 \le i_{1x} \le \mathrm{int}[(M_{Ax} + M_{Bx})/2]$, $0 \le i_{2x} \le \mathrm{int}[(M_{Cx} + M_{Dx})/2]$, $i_1 = i_{1x} + i_{1y} + i_{1z}$, $i_2 = i_{2x} + i_{2y} + i_{2z}$, and:

$$g_x = \binom{M_{Ax} + M_{Bx}}{2i_{1x}} \binom{M_{Cx} + M_{Dx}}{2i_{2x}} (2i_{1x} - 1)!! \times (2i_{2x} - 1)!! (2M_x - 2i_{1x} - 2i_{2x} - 1)!! \quad (8)$$

The molecular incomplete gamma function can be defined as:

$$F_m(z) = \int_0^1 t^{2m} \exp(-zt^2)\, dt \quad (9)$$

in which $z = \overline{PQ}^2/4\delta$, $\delta = 1/4\gamma_1 + 1/4\gamma_2$, $\overrightarrow{PQ} = \mathbf{Q} - \mathbf{P}$, $\mathbf{P} = (\alpha_A \mathbf{A} + \alpha_B \mathbf{B})/\gamma_1$, and $\mathbf{Q} = (\alpha_C \mathbf{C} + \alpha_D \mathbf{D})/\gamma_2$.

In eq. (3), the accompanying coordinate part, $C_4\{\mathbf{N}_3\}$, can be rewritten as:

$$C_4\{\mathbf{N}_3\} = \sum_{ijkl} D_i^{A3} D_j^{B3} D_k^{C3} D_l^{D3} \quad (10)$$

where $i = i'(i' + 1)/2 + i''$ and so on. The corresponding core part $H_4\{\mathbf{N}_3\}$ can be rewritten as $H_{ijkl}(i_A i_B i_C i_D)$ where $i_A = L_A - M_A$, and so on. Then, the ACE-b3k3 general formula, eq. (1), can be rewritten as:

$$\mathrm{ERI} = \sum_{ijkl} D_i^{A3} D_j^{B3} D_k^{C3} D^{D3} H_{ijkl}(i_A i_B i_C i_D) \quad (11)$$

where the factor A_0 is included in the core part H_{ijkl}.

In a previous study,[2] we obtained a general b3k3 algorithm inductively by expanding eq. (11) for individual cases as $(pp|pp)$ and $(dd|dd)$. However, by expanding eq. (11) for more general cases such as $(ff|ff)$, $(gg|gg)$, and $(hh|hh)$, it is found that we must revise the previous algorithm. Hereafter, we use the phrase "it is found that ..." on the findings obtained inductively from the individual expansion. This manipulation is so large that it cannot be shown here. The new ACE-b3k3 general algorithm can be described as follows:

$$\mathrm{ERI} = H_{ABCD}(L_A L_B L_C L_D) \quad (12)$$

The 0th order H_{ABCD} can be defined as:

$$\begin{aligned} H_{ABCD} = {} & \sum_i D_i^{A3}(L_A) H_{iBCD}(L_A L_B L_C L_D) \\ & + \delta_{AA} H_{ABCD}^{AA}(L_A - 2L_B L_C L_D) \\ & + \delta_{AB} H_{ABCD}^{AB}(L_A - 1L_B - 1L_C L_D) \\ & + \delta_{AC} H_{ABCD}^{AC}(L_A - 1L_B L_C - 1L_D) \\ & + \delta_{AD} H_{ABCD}^{AD}(L_A - 1L_B L_C L_D - 1) \end{aligned} \quad (13)$$

where:

$$\delta_{AA} = \delta_{AAx} + {}_{AAy} + \delta_{AAz} \quad (14a)$$

$$\delta_{AAx} = (l_A - M_{Ax})(l_A - M_{Ax} - 1)/2 \quad (14b)$$

$$\delta_{AB} = \delta_{ABx} + \delta_{ABy} + \delta_{ABz} \quad (14c)$$

$$\delta_{ABx} = (l_A - M_{Ax})(l_B - M_{Bx}) \quad (14d)$$

and so on. Eq. (13) is the same as in the previous study,[2] except for eq. (14b). When $l_A \le 2$, eq. (14b) is identical to what was shown in the previous work.[2] The first order H_{ABCD} can be defined as:

$$\begin{aligned} H_{ABCD}^{AA} = {} & \sum_i D_i^{A3}(L_A - 2) H_{iBCD}^{AA}(L_A - 2L_B L_C L_D) \\ & + (1/2)\delta_{AA} H_{ABCD}^{AAAA}(L_A - 4L_B L_C L_D) \\ & + (1/2)\delta_{AB} H_{ABCD}^{AAAB}(L_A - 3L_B - 1L_C L_D) \\ & + (1/2)\delta_{AC} H_{ABCD}^{AAAC}(L_A - 3L_B L_C - 1L_D) \\ & + (1/2)\delta_{AD} H_{ABCD}^{AAAD}(L_A - 3L_B L_C L_D - 1) \end{aligned} \quad (15)$$

and so on. The second order H_{ABCD} can be defined as:

$$\begin{aligned} H_{ABCD}^{AAAA} + \sum_i {} & D_i^{A3}(L_A - 4) H_{iBCD}^{AAAA}(L_A - 4L_B L_C L_D) \\ & + (1/3)\delta_{AA} H_{ABCD}^{AAAAAA}(L_A - 6L_B L_C L_D) \\ & + (1/3)\delta_{AB} H_{ABCD}^{AAAAAB}(L_A - 5L_B - 1L_C L_D) \\ & + (1/3)\delta_{AC} H_{ABCD}^{AAAAAC}(L_A - 5L_B L_C - 1L_D) \\ & + (1/3)\delta_{AD} H_{ABCD}^{AAAAAD}(L_A - 5L_B L_C L_D - 1) \end{aligned} \quad (16)$$

and so on. Eqs. (15) and (16) are new. It is found that $H_{ABCD}^{AAAB} = H_{ABCD}^{ABAA}$, $H_{ABCD}^{ABAC} = H_{ABCD}^{ACAB}$, and so on. Thus, the second order $H_{ABCD}(i_A i_B i_C i_D)$ depends on the combination of these pairs of AA, AB, AC, and AD. The combination of these pairs can be sufficiently determined by the four indices

$(i_A i_B i_C i_D)$ (where $i_A = L_A - M_A$, and so on); therefore, it is redundant at this stage to use the pair notation of the superscript with $(i_A i_B i_C i_D)$. The reason for using the pair notation is clarified later. Hereafter, we use the dictionary order for representing these pairs; that is, we use $ABAC$ and not $ACAB$, and so on, at the superscript of H_{ABCD}. The higher order H_{ABCD} can be defined in the same manner as done previously. In the definition of the ith order H_{ABCD}, the factor before δ_{AA} (δ_{AB}, δ_{AC}, or δ_{AD}) is $1/(i+1)$. The chain definition of H_{ABCD}, eqs. (13)–(16), is closed because $H^*_{ABCD}(i_A i_B i_C i_D) = 0$ when there is a negative value among i_A, i_B, i_C, and i_D. Hereafter, superscript * denotes any set of pairs.

The 0th order H_{iBCD} can be defined as:

$$\begin{aligned} H_{iBCD}(L_A L_B L_C L_D) &= \sum_j D_j^{B3}(L_B) H_{ijCD}(L_A L_B L_C L_D) \\ &+ \delta_{BB} H_{iBCD}^{BB}(L_A L_B - 2L_C L_D) \\ &+ \delta_{BC} H_{iBCD}^{BC}(L_A L_B - 1L_C - 1L_D) \\ &+ \delta_{BD} H_{iBCD}^{BD}(L_A L_B - 1L_C L_D - 1) \end{aligned} \quad (17)$$

where δ_{BB}, δ_{BC}, and δ_{BD} can be expressed as in eq. (14). The other type of the 0th order H_{iBCD} can be defined as:

$$\begin{aligned} H_{iBCD}^{AA}(L_A - 2L_B L_C L_D) &= \sum_j D_j^{B3}(L_B) H_{ijCD}^{AA}(L_A - 2L_B L_C L_D) \\ &+ \delta_{BB} H_{iBCD}^{AABB}(L_A - 2L_B - 2L_C L_D) \\ &+ \delta_{BC} H_{iBCD}^{AABC}(L_A - 2L_B - 1L_C - 1L_D) \\ &+ \delta_{BD} H_{iBCD}^{AABD}(L_A - 2L_B - 1L_C L_D - 1) \end{aligned} \quad (18)$$

and so on. Each 0th order H^*_{iBCD} corresponds to each of order of H^*_{ABCD}, given in eqs. (13)–(16). Eqs. (17) and (18) are the same as in the previous work[2] except for the value of δ_{BB}. The first order H_{iBCD} can be defined as:

$$\begin{aligned} H_{iBCD}^{BB}(L_A L_B - 2L_C L_D) &= \sum_j D_j^{B3}(L_B - 2) H_{ijCD}^{BB}(L_A L_B - 2L_C L_D) \\ &+ (1/2)\delta_{BB} H_{iBCD}^{BBBB}(L_A L_B - 4L_C L_D) \\ &+ (1/2)\delta_{BC} H_{iBCD}^{BBBC}(L_A L_B - 3L_C - 1L_D) \\ &+ (1/2)\delta_{BD} H_{iBCD}^{BBBD}(L_A L_B - 3L_C L_D - 1) \end{aligned} \quad (19)$$

$$\begin{aligned} H_{iBCD}^{AABB}(L_A - 2L_B - 2L_C L_D) &= \sum_j D_j^{B3}(L_B - 2) H_{ijCD}^{AABB}(L_A - 2L_B - 2L_C L_D) \\ &+ (1/2)\delta_{BB} H_{iBCD}^{AABBBB}(L_A - 2L_B - 4L_C L_D) \\ &+ (1/2)\delta_{BC} H_{iBCD}^{AABBBC}(L_A - 2L_B - 3L_C - 1L_D) \\ &+ (1/2)\delta_{BD} H_{iBCD}^{AABBBD}(L_A - 2L_B - 3L_C L_D - 1) \end{aligned} \quad (20)$$

and so on. It is found that $H_{iBCD}^{BCBD} = H_{iBCD}^{BDBC}$, and so on. Thus, $H^*_{iBCD}(i_A i_B i_C i_D)$ depends on the combination of pairs of AA, AB, AC, AD, BB, BC, and BD. Hereafter, we use the dictionary order for representing these pairs; that is, we use $ABBBBC$ and not $ABBCBB$, and so on. It is also found that $H_{iBCD}^{ABAB} \neq H_{iBCD}^{AABB}$ and $H_{iBCD}^{ABAC} \neq H_{iBCD}^{AABC}$, because H_{iBCD}^{ABAB} (or H_{iBCD}^{ABAC}) is the 0th order one, whereas H_{iBCD}^{AABB} (or H_{iBCD}^{AABC}) is first order. This is the reason to use pair notation at the superscript. When $L_B \leq 2$, however, it is found that $H_{iBCD}^{ABAB} = H_{iBCD}^{AABB}$ and $H_{iBCD}^{ABAC} = H_{iBCD}^{AABC}$. For this case, the pair notation is not necessary, because $(i_A i_B i_C i_D)$ of H^*_{iBCD} can sufficiently describe the term. This is the situation for description of the ACE-b3k3 algorithm in the previous work.[2] For $L_B \geq 3$, we must use the pair notation. The pair notation can completely determine the values of i_A, i_B, i_C, and i_D. Therefore, the argument $(i_A i_B i_C i_D)$ is not necessary in the expression. It is mainly for readers' convenience. Higher order H^*_{iBCD} can be defined in the same way as in eqs. (19) and (20). The chain definition of H^*_{iBCD} is closed because $H^*_{iBCD}(i_A i_B i_C i_D) = 0$ when there is a negative value among i_A, i_B, i_C, and i_D.

Throughout this paper, we use the term "ith order" for H^*_{ABCD}, and H^*_{iBCD} (H^*_{ijCD} and H^*_{ijkD} also appear later) when these have the factor of $1/(i+1)$ before δ_{AA} (or δ_{BB}, and so on). This factor is obtained inductively from expanding eq. (11) for individual cases.

The 0th order H^*_{ijCD} can be defined as:

$$\begin{aligned} H^*_{ijCD}(i_A i_B i_C i_D) &= \sum_k D_k^{C3}(i_C) H^*_{ijkD}(i_A i_B i_C i_D) \\ &+ \delta_{CC} H^{*CC}_{ijCD}(i_A i_B i_C - 2i_D) \\ &+ \delta_{CD} H^{*CD}_{ijCD}(i_A i_B i_C - 1i_D - 1) \end{aligned} \quad (21)$$

Higher order H^*_{ijCD} can be defined in a similar way as in H^*_{iBCD}. It is found that $H_{ijCD}^{ABAB} = H_{ijCD}^{AABB}$

and $H_{ijCD}^{ABAC} = H_{ijCD}^{AABC}$ in all cases. Then the total number of 0th order H_{ijCD}^{*} is less than the total number of all orders of H_{iBCD}^{*}. For $L_C \le 2$, it is found that the four values of $(i_A i_B i_C i_D)$ can sufficiently describe the term of all orders of H_{ijCD}^{*}. This is the case in the previous work.[2]

The 0th order H_{ijkD}^{*} can be defined as:

$$H_{ijkD}^{*}(i_A i_B i_C i_D) = \sum_l D_l^{D3}(i_D) H_{ijkl}(i_A i_B i_C i_D) + \delta_{DD} H_{ijkD}^{*DD}(i_A i_B i_C i_D - 2) \quad (22)$$

It is found that H_{ijkl} can be described sufficiently by the four indices $(i_A i_B i_C i_D)$; therefore, the pair notation at the superscript is always omitted for H_{ijkl}. Higher order H_{ijkD}^{*} can be also defined in a similar way as in H_{iBCD}^{*}. It is found that $H_{ijkD}^{ABAB} = H_{ijkD}^{AABB}$ and $H_{ijkD}^{ABAC} = H_{ijkD}^{AABC}$ in all cases. The total number of 0th order H_{ijkD}^{*} is equal to the total number of all orders of H_{ijCD}^{*}. It is found that $H_{ijkD}^{ADCD} \neq H_{ijkD}^{ACDD}$ and $H_{ijkD}^{CDCD} \neq H_{ijkD}^{CCDD}$, because H_{ijkD}^{ADCD} (or H_{ijkD}^{CDCD}) is 0th order, whereas H_{ijkD}^{ACDD} (or H_{ijkD}^{CCDD}) is first order.

For $L_D \le 2$, however, $H_{ijkD}^{ADCD} = H_{ijkD}^{ACDD}$ and $H_{ijkD}^{CDCD} = H_{ijkD}^{CCDD}$. This was the case in the previous study.[2] All calculations of H_{ijkl} are also exactly the same as in the previous study.[2] Thus, with no loss of generality, we can assume that the innermost contraction loop (so-called K^4 loop) is the K_{bra} loop (when the innermost is K_{ket}, then the following procedure should be read after exchanging K_{bra} for K_{ket}).

Let us define F_{mn} and G_{mn}^{pq} by:

$$F_{mn} = \left(\frac{\sigma_1}{\sigma_1 + \sigma_2}\right)^m \left(\frac{\sigma_2}{\sigma_1 + \sigma_2}\right)^n F_{m+n}(z)$$

where $0 \le m \le L_A + L_B$, $0 \le n \le L_C + L_D$, and:

$$G_{mn}^{pq} = \sigma_A^p \sigma_B^q \sum_{t=0}^{\mu} (-)^t \binom{\mu}{t} F_{m+t\,n}$$

where $\mu = L_A + L_B - m$ and $0 \le p, q \le n + \mu$ with $p + q = n + \mu$.

The computation of F_{mn} and G_{mn}^{pq} must be performed at the K^4 loop. The other necessary term at the K^4 loop is:

$$g_{m\lambda_1}^{pq} = \sigma_A^p \sigma_B^q \sigma_1^{\lambda_1} \sum_{t=0}^{\mu} (-)^t \binom{\mu}{t} F_{m-\lambda_1+t\,0}$$

where $1 \le \lambda_1 \le \min\{m, \mu, \mathrm{int}[(L_A + L_B)/2]\}$ and $0 \le p, q \le \mu - \lambda_1$ with $p + q = \mu - \lambda_1$. Let us define H_{mn}^{pqrs} by:

$$H_{mn}^{pqrs} = \sigma_C^r \sigma_D^s \sum_{t=0}^{\nu} (-)^t \binom{\nu}{t} G_{m\,n+t}^{pq}$$

where $\nu = L_C + L_D - n$, $\mu \le p + q \le n + \mu$, and $\nu \le r + s \le m + \nu$.

The computation of H_{mn}^{pqrs} can be performed at the K_{ket} loop (so-called K^2 loop) for $p + q = n + \mu$ and $r + s = m + \nu$, and can be done at out of the contraction loops (so-called K^0 step) for $\mu \le p + q < m + \mu$ and $\nu \le r + s < m + \nu$ by the use of the following relations:

$$H_{mn}^{pqrs} = H_{mn}^{p+1\,qrs} + H_{mn}^{p\;q+1\,rs} \quad (\mu \le p + q < n + \mu)$$

and:

$$H_{mn}^{pqrs} = H_{mn}^{pq\,r+1\,s} + H_{mn}^{pq\,r\,s+1} \quad (\nu \le r + s < m + \nu)$$

These relations are based on $\sigma_A + \sigma_B = 1$ and $\sigma_C + \sigma_D = 1$. The other necessary term at the K^2 loop is:

$$h_{mn\lambda_2}^{pqrs} = \sigma_C^r \sigma_D^s \sigma_2^{\lambda_2} \sum_{t=0}^{\nu} (-)^t \binom{\nu}{t} G_{m\,n-\lambda_2+t}^{pq}$$

where $1 \le \lambda_2 = (M_A + M_B + M_C + M_D)/2 \le \min\{m + \nu, n + \mu, m + n - \mathrm{int}[(L_A + L_B + L_C + L_D)/2]\}$, $\max\{\mu - \lambda_2, 0\} \le p + q \le n + \mu - \lambda_2$, and $\max\{\nu - \lambda_2, 0\} \le r + s \le m + \nu - \lambda_2$. We must use $g_{m\lambda_1}^{pq}$ instead $\sigma_2^{\lambda_1} G_{m-\lambda_1}^{pq}$ when $n - \lambda_2 + t$ is negative in the above equation with $\lambda_1 = -(n - \lambda_2 + t)$. The computation of $h_{mn\lambda_2}^{pqrs}$ can be performed at the K^2 loop for $p + q = n + \mu - \lambda_2$ and $r + s = m + \nu - \lambda_2$ and can be done at the K^0 step for $p + q < n - \mu - \lambda_2$ and $r + s < m + \nu - \lambda_2$ with the use of the following relations:

$$h_{mn\lambda_2}^{pqrs} = h_{mn\lambda_2}^{p+1\,qrs} + h_{mn\lambda_2}^{p\;q+1rs} \quad (\max\{\mu - \lambda_2, 0\} \le p + q < n + \mu - \lambda_2)$$

$$h_{mn\lambda_2}^{pqrs} = h_{mn\lambda_2}^{pq\;r+1s} + h_{mn\lambda_2}^{pq\;rs+1} \quad (\max\{\nu - \lambda_2, 0\} \le r + s < m + \nu - \lambda_2)$$

These relations are based on $\sigma_A + \sigma_B = 1$ and $\sigma_C + \sigma_D = 1$. All necessary H_{ijkl} can be computed from the above terms by the use of eq. (6).

Finally, the abstract of the new ACE-b3k3 general algorithm is shown in Figure 1. In the Figure 1, the four loop indices $(i_A i_2 i_3 i_4)$ describe the "A pairs"; for example, $AXAX$, where $X = A$, B, C, and D, and so on ($i_A = 2$ $i_1 = 1$ $i_3 = 1$ $i_4 = 0$ for

```
loop over K_ket
 loop over K_bra
  compute G^pq_mn and g^pq_m λ1                    (K^4 step)
 end loop K_bra
 compute H^pqrs_mn and h^pqrs_mnλ2                 (K^2 step)
end loop K_ket
(K^0 step)
loop over i_A=L_A to 0 step -1
 i_2max=min{i_A, L_B}
loop over i_2=i_2max to 0 step -1
 i_3max=min{i_A-i_2, L_C}
loop over i_3=i_3max to 0 step -1
 i_4max=min{i_A-i_2-i_3, L_D}
loop over i_4=i_4max to 0 step -1
 if i_A+i_2+i_3+i_4≠even then skip this loop
 loop over j_1=L_B-i_2 to 0 step -1
  j_2max=min{j_1, L_C-i_3}
 loop over j_2=j_2max to 0 step -1
  j_3max=min{j_1-j_2, L_D-i_4}
 loop over j_3=j_3max to 0 step -1
  if j_1+j_2+j_3≠even then skip this loop
  loop over k_1=L_C-i_3-j_2 to 0 step -1
   k_2max=min{k_1, L_D-i_4-j_3}
  loop over k_2=k_2max to 0 step -1
   if k_1+k_2≠even then skip this loop
```

FIGURE 1. Abstract of the ACE-b3k3 general algorithm.

ABAC). The three loop indices ($j_1\, j_2\, j_3$) describe the ''*B* pairs''; For example, *BXBX*, where $X = B$, *C*, and *D*, and so on ($j_1 = 4$ $j_2 = 1$ $j_3 = 1$ for *BBBCBD*).

The two loop indices (k_1 k_2) describe the ''*C* pairs''; for example, *CXCXCX*, where $X = C$ and *D*, and so on ($k_1 = 3$ $k_2 = 1$ for *CCCD*). The loop index m_1 describes the total number of the *DD* pairs.

For the reader's convenience, in order to observe the present algorithm, the expanded formula of eq. (11) for the ($pp|pp$) class of ERIs is shown. The expanded formula of those of higher order GTOs is too extensive to be shown here. Thus:

$$
\begin{aligned}
&\left(p_i\, p_j|\, p_k\, p_l\right) \\
&\quad = AB_i\Big[\, BA_j\big\{CD_k(\,DC_l H_{00}^{1111} \\
&\qquad + BA_l H_{01}^{2101} + DB_l H_{01}^{1101}) \\
&\qquad + AB_k(\,DC_l H_{01}^{1210} + BA_l H_{02}^{2200} + DB_l H_{02}^{1200}) \\
&\qquad + CA_k(\,DC_l H_{01}^{1110} + BA_l H_{02}^{2100} \\
&\qquad + DB_l H_{02}^{1100}) + \delta_{kl} h_{011}^{1100}\big\} \\
&\qquad + DC_j\big\{CD_k(\,DC_l H_{10}^{0121}
\end{aligned}
$$

```
    if L_D-i_4-j_3-k_2≥2 then
    loop over m_1=int[(L_D-i_4-j_3-k_2)/2] to 1 step -1
     compute H_ijkℓ(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2-k_1 L_D-i_4-j_3-k_2-2m_1)
     compute H*_ijkD(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2-k_1 L_D-i_4-j_3-k_2-2m_1)
    end loop m_1
    end if
    compute H_ijkℓ(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2-k_1 L_D-i_4-j_3-k_2)
    compute H*_ijkD(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2-k_1 L_D-i_4-j_3-k_2)
    if k_1>0 then compute
            H*_ijCD(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2-k_1 L_D-i_4-j_3-k_2)
    end loop k_2
    end loop k_1
    compute H*_ijCD(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2 L_D-i_4-j_3)
    if j_1>0 then compute
            H*_iBCD(L_A-i_A L_B-i_2-j_1 L_C-i_3-j_2 L_D-i_4-j_3)
   end loop j_3
   end loop j_2
   end loop j_1
   compute H*_iBCD(L_A-i_A L_B-i_2 L_C-i_3 L_D-i_4)
   if i_A>0 then compute H*_ABCD(L_A-i_A L_B-i_2 L_C-i_3 L_D-i_4)
  end loop i_4
  end loop i_3
  end loop i_2
  end loop i_A
 compute H_ABCD(L_A L_B L_C L_D)
```

FIGURE 1. (Continued)

$$
\begin{aligned}
&+BA_l H_{11}^{1111} + DB_l H_{11}^{0111})\\
&+AB_k(DC_l H_{11}^{0220} + BA_l H_{12}^{1210} + DB_l H_{12}^{0210})\\
&+CA_k(DC_l H_{11}^{0120} + BA_l H_{12}^{1110}\\
&+DB_l H_{12}^{0110}) + \delta_{kl} h_{111}^{0110}\}\\
&+BD_j\{CD_k(DC_l H_{10}^{0111} + BA_l H_{11}^{1101}\\
&+DB_l H_{11}^{0101})\\
&+AB_k(DC_l H_{11}^{0210} + BA_l H_{12}^{1200} + DB_l H_{12}^{0200})\\
&+CA_k(DC_l H_{11}^{0110} + BA_l H_{12}^{1100}\\
&+DB_l H_{12}^{0100}) + \delta_{kl} h_{111}^{0100}\}\\
&+\delta_{jk}(DC_l h_{111}^{0110} + BA_l h_{121}^{1100} + DB_l h_{121}^{0100})\\
&+\delta_{jl}(CD_k h_{111}^{0101} + AB_k h_{121}^{0200} + CA_k h_{121}^{0100})\big]\\
&+CD_i\big[BA_j\{CD_k(DC_l H_{10}^{1012}\\
&+BA_l H_{11}^{2002} + DB_l H_{11}^{1002})\\
&+AB_k(DC_l H_{11}^{1111} + BA_l H_{12}^{2101} + DB_l H_{12}^{1101})\\
&+CA_k(DC_l H_{11}^{1011} + BA_l H_{12}^{2001}\\
&+DB_l H_{12}^{1001}) + \delta_{kl} h_{111}^{1001}\}\\
&+DC_j\{CD_k(DC_l H_{20}^{0022} + BA_l H_{21}^{1012}
\end{aligned}
$$

$$
\begin{aligned}
&+ DB_l H_{21}^{0012}) \\
&+ AB_k (DC_l H_{21}^{0121} + BA_l H_{22}^{1110} + DB_l H_{22}^{0110}) \\
&+ CA_k (DC_l H_{21}^{0021} + BA_l H_{22}^{1011} \\
&+ DB_l H_{22}^{0011}) + \delta_{kl} h_{211}^{0011}\} \\
&+ BD_j \{CD_k (DC_l H_{20}^{0012} + BA_l H_{21}^{1002} \\
&+ DB_l H_{21}^{0002}) \\
&+ AB_k (DC_l H_{21}^{0111} + BA_l H_{22}^{1101} + DB_l H_{22}^{0101}) \\
&+ CA_k (DC_l H_{21}^{0011} + BA_l H_{22}^{1001} \\
&+ DB_l H_{22}^{0001}) + \delta_{kl} h_{211}^{0001}\} \\
&+ \delta_{jk} (DC_l h_{211}^{0011} + BA_l h_{221}^{1001} + DB_l h_{221}^{0001}) \\
&+ \delta_{jl} (CD_k h_{211}^{0002} + AB_k h_{221}^{0101} + CA_k h_{221}^{0001})\Big] \\
&+ AC_i \Big[BA_j \{CD_k (DC_l H_{10}^{1011} \\
&+ BA_l H_{11}^{2001} + DB_l H_{11}^{1001}) \\
&+ AB_k (DC_l H_{11}^{1110} + BA_l H_{12}^{2100} + DB_l H_{12}^{1100}) \\
&+ CA_k (DC_l H_{11}^{1010} + BA_l H_{12}^{2000} \\
&+ DB_l H_{12}^{1000}) + \delta_{kl} h_{111}^{1000}\} \\
&+ DC_j \{CD_k (DC_l H_{20}^{0021} + BA_l H_{21}^{1011} \\
&+ DB_l H_{21}^{0011}) \\
&+ AB_k (DC_l H_{21}^{0120} + BA_l H_{22}^{1110} + DB_l H_{22}^{0110}) \\
&+ CA_k (DC_l H_{21}^{0020} + BA_l H_{22}^{1010} \\
&+ DB_l H_{22}^{0010}) + \delta_{kl} h_{211}^{0010}\} \\
&+ BD_j \{CD_k (DC_l H_{20}^{0011} + BA_l H_{21}^{1001} \\
&+ DB_l H_{21}^{0001}) \\
&+ AB_k (DC_l H_{21}^{0110} + BA_l H_{22}^{1100} + DB_l H_{22}^{0100}) \\
&+ CA_k (DC_l H_{21}^{0010} + BA_l H_{22}^{1000} \\
&+ DB_l H_{22}^{0000}) + \delta_{kl} h_{211}^{0000}\} \\
&+ \delta_{jk} (DC_l h_{211}^{0010} + BA_l h_{221}^{1000} + DB_l h_{221}^{0000}) \\
&+ \delta_{jl} (CD_k h_{211}^{0001} + AB_k h_{221}^{0100} + CA_k h_{221}^{0000})\Big] \\
&+ \delta_{ij} \Big[CD_k (DC_l h_{101}^{0011} + BA_l h_{111}^{1001} + DB_l h_{111}^{0001}) \\
&+ AB_k (DC_l h_{111}^{0110} + BA_l h_{121}^{1100} + DB_l h_{121}^{0100}) \\
&+ CA_k (DC_l h_{111}^{0010} + BA_l h_{121}^{1000} \\
&+ DB_l h_{121}^{0000}) + \delta_{kl} h_{112}^{0000}\Big] \\
&+ \delta_{ik} \Big[BA_j (DC_l h_{111}^{1010} + BA_l h_{121}^{2000} + DB_l h_{121}^{1000}) \\
&+ DC_j (DC_l h_{211}^{0020} + BA_l h_{221}^{1010} + DB_l h_{221}^{0010}) \\
&+ BD_j (DC_l h_{211}^{0010} + BA_l h_{221}^{1000} \\
&+ DB_l h_{221}^{0000}) + \delta_{jl} h_{222}^{0000}\Big] \\
&+ \delta_{il} \Big[BA_j (CD_k h_{111}^{1001} + AB_k h_{121}^{1100} + CA_k h_{121}^{1000}) \\
&+ DC_j (CD_k h_{211}^{0011} + AB_k h_{221}^{0110} + CA_k h_{221}^{0010}) \\
&+ BD_j (CD_k h_{211}^{0001} + AB_k h_{221}^{0100} \\
&+ CA_k h_{221}^{0000}) + \delta_{jk} h_{222}^{0000}\Big]
\end{aligned}
$$

where i, j, k, $l = x$, y, z; δ_{ij} is the Kronecker's delta; $AB_i = B_i - A_i$, and so on. In the above formula, the correspondence between terms is as follows:

$$
\begin{aligned}
AB_i, CD_i, AC_i &\leftrightarrow D_i^{A3}(1) \\
BA_j, DC_j, BD_j &\leftrightarrow D_j^{B3}(1) \\
CD_k, AB_k, CA_k &\leftrightarrow D_k^{C3}(1) \\
DC_l, BA_l, DB_l &\leftrightarrow D_l^{D3}(1) \\
\delta_{ij}, \delta_{ik}, \delta_{il} &\leftrightarrow \delta_{AB}, \delta_{AC}, \delta_{AD} \\
\delta_{jk}, \delta_{jl}, \delta_{kl} &\leftrightarrow \delta_{BC}, \delta_{BD}, \delta_{CD}
\end{aligned}
$$

and:

$$
H_{mn}^{pqrs}, h_{mn\,\lambda_2}^{pqrs} \leftrightarrow H_{ijkl}(i_A i_B i_C i_D)
$$

The present ACE-b3k3 algorithm is especially suitable for general contraction of the basis set; that is, for the atomic natural orbital (ANO) basis of Almlöf and Taylor.[12,13] They used $K = 6$ for the d-type ANO, $K = 4$ for f-type in the Ne or N atom, and $K = 10$ for the d-type in the S atom.[12] The ANO is a high-quality basis set that allows for essentially no loss of correlation energy.[13] By the ACE-b3k3 algorithm, we can use the ANO basis of larger contraction for higher GTOs. For example, we may use $K = 6$ for f-type ANO bases, because the FLOP count of $K = 6$ is comparable to that of $k = 3$–5 (see Table VI). A special code for the ANO basis can be made, which is a project currently in progress.

The present algorithm can be also applied to the derivative of ERI. Such a project is also in progress.

Numerical Examples

A computer program code that is universal for all types of GTOs can be constructed by the use of the present ACE-b3k3 general algorithm described in the preceding section. Table I shows the FLOP count parameters of the present version of the general program code for $(LL|LL)$ ERIs ($L = 1$–5). The value of the x parameter is the FLOP count of

TABLE I.
The FLOP Count Parameters of ACE-b3k3 General Algorithm.[a,b]

ERI class	x	y	z
$(pp\|pp)$	75 (61)[c]	205 (166)	2318 (1920)
$(dd\|dd)$	327	2281	163000
$(ff\|ff)$	861	11237	4146000
$(gg\|gg)$	1781	37128	58022000
$(hh\|hh)$	3191	96749	538709000

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]FLOP count $= xK^4 + yK^2 + z$, where K is the degree of contraction.
[c]Parameters of the computer program code, specially optimized for the $(pp|pp)$ ERI class, are shown in parentheses (see ref. 2).

the so-called K^4 step. The value of the y parameter is that of K^2 step. The value of the z parameter is that of the K^0 step. The total FLOP count is then given by:

$$\text{FLOP count} = xK^4 + yK^2 + z \qquad (23)$$

where K is the degree of contraction of GTO. In eq. (23), it is assumed that all GTOs in the ERI in question have the same K value. The FLOP count to make the molecular incomplete gamma function $F_m(z)$ is omitted in this article, but has been investigated in the previous work.[2] The FLOP count in Table I is not optimal for two reasons: (1) the computer program code is universal for all types of GTOs; and (2) this code is not yet fully optimized in its present version. For example, for $(pp|pp)$ ERIs, the parameter values are $x = 75$, $y = 205$, $z = 2318$, whereas the optimum values are $x = 61$, $y = 166$, $z = 1920$, as shown in the previous article.[2] However, the present version of the program code is practically sufficient for d or higher-order GTOs as discussed in what follows.

Tables II–IV show comparisons to other methods in FLOP count parameters. The methods of Ishida,[3] Head-Gordon and Pople,[4] Lindh et al.,[5] and Hamilton and Schaefer[6] are comparable in regard to FLOP count parameters, as seen in Tables II and III.

For a vector computer, Ishida's method II can be strongly recommended among these comparable methods, because of the strongly vectorizable feature in the K^4 step.[3] The methods of McMurchie and Davidson[7] and Dupuis et al.[8] have very large x parameters, which means that these methods must not be used for larger K. The PRISM algorithm of Gill and Pople[9] and the present ACE-b3k3 have small x parameters, which means that they are adequate for larger K. As seen in Tables II–IV, the x parameter of the ACE-b3k3 is the lowest among all methods in the literature.

The total FLOP count of the present ACE-b3k3 is compared with the other methods in Tables

TABLE II.
Comparison of FLOP Count Parameters with Those of Other Methods for $(dd|dd)$ ERI Class.[a,b]

Method	x	y	z
ACE-b3k3	327	2281	163000
Gill and Pople (PRISM-CCTTT)[c]	575	5506	159624
Tenno's estimate[d]	330	1800	200000
Ishida's method II[e]	13209	30	11256
Head-Gordon and Pople[c]	13466	0	10295
Lindh et al.[f]	10255	30	11256
Hamilton and Schaefer[g]	13900	30	11256
McMurchie and Davidson[g]	27300	24000	0
Dupuis et al.[g]	30900	220	0

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]FLOP count $= xK^4 + yK^2 + z$, where K is the degree of contraction.
[c]Ref. 9.
[d]Ref. 10.
[e]Ref. 3.
[f]Ref. 5.
[g]Ref. 6.

TABLE III.
Comparison of FLOP Count Parameters with Those of Other Methods for $(ff|ff)$ ERI class.[a,b]

Method	x	y	z
ACE-b3k3	861	11237	4146000
Gill et al.[c]	11000	600000	600000
Ishida's method II[d]	77817	30	135024
Head-Gordon and Pople[e]	108000	30	135024
Lindh et al.[f]	76901	30	135024
Hamilton and Schaefer[g]	87800	30	135024
McMurchie and Davidson[g]	342000	383000	0
Dupuis et al.[g]	276000	600	0

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]FLOP count $= xK^4 + yK^2 + z$, where K is the degree of contraction.
[c]Ref. 11.
[d]Ref. 3.
[e]Ref. 4.
[f]Ref. 5.
[g]Ref. 6.

TABLE IV.
Comparison of FLOP Count Parameters with Those of Ishida's Method II (Ref. 3) for (*gg*|*gg*) and (*hh*|*hh*) ERI Classes.[a,b]

Method	*x*	*y*	*z*
(*gg*|*gg*)			
ACE-b3k3	1781	37128	58022000
Ishida's method II	308615	30	939060
(*hh*|*hh*)			
ACE-b3k3	3191	96749	538709000
Ishida's method II	951864	30	4651624

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]FLOP count = $xK^4 + yK^2 + z$, where K is the degree of contraction.

V–VII for (*dd*|*dd*), (*ff*|*ff*), (*gg*|*gg*), and (*hh*|*hh*) ERIs. The methods of McMurchie and Davidson[7] and Dupuis et al.[8] must not be used when $K \geq 3$, as seen in Tables V and VI. Ishida's[3] method II and comparable methods can be used when $K \leq 3$ for (*ff*|*ff*), $K \leq 4$ for (*gg*|*gg*), and $K \leq 5$ for (*hh*|*hh*). The present ACE-b3k3 is fastest when $K \geq 3$ for (*dd*|*dd*) and for (*ff*|*ff*), $K \geq 4$ for (*gg*|*gg*), and $K \geq 5$ for (*hh*|*hh*), as seen in Tables V–VII. Ishida's method II is fastest when $K = 3$ for (*gg*|*gg*) and $3 \leq K \leq 4$ for (*hh*|*hh*).

Table VIII shows the total computation time of ERI by the present ACE-b3k3 general code compared with Ishida's[3] method II for (*LL*|*LL*) ERIs ($L = 2$–5) measured with a HITAC-MP5800 scalar

TABLE V.
Comparison of Total FLOP Count with that of Other Methods for (*dd*|*dd*) ERI Class.[a,b]

Method	$K = 3$	$K = 4$	$K = 5$	$K = 6$
ACE-b3k3	210000	283000	424000	669000
Gill and Pople (PRISM)[c]	256000	395000	657000	1103000
Tenno's estimate[d]	243000	313000	451000	692000
Ishida's method II[e]	1081000	3393000	8268000	17131000
Head-Gordon and Pople[c]	1101000	3458000	8427000	17462000
Lindh et al.[f]	842000	2637000	6421000	13303000
Hamilton and Schaefer[g]	1137000	3570000	8700000	18027000
McMurchie and Davidson[g]	2427000	7373000	17663000	36245000
Dupuis et al.[g]	2505000	7914000	19318000	40054000

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]K is the degree of contraction.
[c]Ref. 9.
[d]Ref. 10.
[e]Ref. 3.
[f]Ref. 5.
[g]Ref. 6.

TABLE VI.
Comparison of Total FLOP Count with that of Other Methods for (*ff*|*ff*) ERI Class.[a,b]

Method	$K = 3$	$K = 4$	$K = 5$	$K = 6$
ACE-b3k3	4317000	4547000	4966000	5666000
Gill et al.[c]	6891000	13016000	22475000	36456000
Ishida's method II[d]	6438000	20057000	48771000	100987000
Head-Gordon and Pople[e]	8883000	27783000	67636000	140104000
Lindh et al.[f]	6372000	19822000	48199000	99800000
Hamilton and Schaefer[g]	7247000	22612000	55011000	113925000
McMurchie and Davidson[g]	31149000	93680000	223325000	457020000
Dupuis et al.[g]	22356000	70666000	172515000	357718000

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]K is the degree of contraction.
[c]Ref. 11.
[d]Ref. 3.
[e]Ref. 4.
[f]Ref. 5.
[g]Ref. 6.

TABLE VII.
Comparison of Total FLOP Count with that of Ishida's Method II (Ref. 3) for ($gg|gg$) and ($hh|hh$) ERI Classes.[a,b]

Method	$K = 3$	$K = 4$	$K = 5$	$K = 6$
($gg|gg$)				
ACE-b3k3	58501000	59073000	60064000	61667000
Ishida's method II	25937000	79945000	193824000	400905000
($hh|hh$)				
ACE-b3k3	539838000	541073000	543122000	546198000
Ishida's method II	81753000	248329000	599567000	1238268000

[a]Except for the calculation of $F_m(z)$; see ref. 2 for $F_m(z)$.
[b]K is the degree of contraction.

TABLE VIII.
Total Computation Time of Class of ERIs by ACE-b3k3 Algorithm Compared with that Using Ishida's Method II (Ref. 3) (in Seconds).[a,b]

ERI class	$K = 3$	$K = 4$	$K = 5$	$K = 6$
($dd|dd$)	0.0096	0.0137	0.0217	0.0356
	0.0436	0.138	0.338	0.705
($ff|ff$)	0.132	0.143	0.162	0.196
	0.266	0.835	2.04	4.27
($gg|gg$)	1.70	1.73	1.77	1.84
	1.14	3.52	8.80	18.1
($hh|hh$)	16.3	16.3	16.4	16.5
	3.76	11.6	28.3	58.3

[a]ACE-b3k3 times are upper entries and Ishida's method II are lower entries. All times were measured with a HITAC MP5800 scalar computer.
[b]K is the degree of contraction.

computer. For ($dd|dd$) and ($ff|ff$), the ACE-b3k3 is 2−20 times faster than Ishida's method II, as seen in Table VIII. For ($gg|gg$), the ACE-b3k3 is 2−20 times faster for $K \geq 4$. For ($hh|hh$), it is 1.5−3 times faster for $K \geq 5$. For a smaller value of K, formulas other than ACE b3k3 must be used (e.g., b2k3 or b1k1).

Construction of the computer program code for b2k3 and b1k1, which is universal for all types of GTOs, is in progress.

It is confirmed with the present general program code that the ACE-b3k3 algorithm is numerically very stable even for higher order GTOs; that is, there is no loss of significant figures other than the round-off error. For ($gg|gg$) and ($hh|hh$), the total FLOP count of the ACE-b3k3 is varied very slightly along increasing K value, as seen in Table VII. This is because the value of the z parameter is very large. For the higher order GTOs d−h, it is more efficient to use the solid harmonic (SH) GTOs, given by:

$$\text{SH-GTO} = S_{lm}\exp(-\alpha r^2) \tag{24}$$

where the solid harmonics S_{lm} of a polar coordinate (r, θ, ϕ) can be defined as:

$$S_{l0} = r^l P_l(\cos\theta) \tag{25a}$$

$$S_{lm} = r^l P_{lm}(\cos\theta)\cos m\phi \quad (m > 0) \tag{25b}$$

and:

$$S_{l-m} = r^l P_{lm}(\cos\theta)\sin m\phi \quad (m > 0) \tag{25c}$$

For SH-GTOs, the value of the z parameter (K^0 FLOP count) will be drastically smaller than that for the usual Cartesian GTO. This will permit drastically faster computation of ERI for higher order GTOs. Such a project is currently in progress.

Acknowledgments

All computations were carried out using a HITAC MP5800 computer at the Computer Centre of the University of Tokyo. The author thanks his wife for her assistance in typing his program code into the computer. The author thanks the referee for many helpful comments.

References

1. For example, see S. Ten-no and S. Iwata, *J. Chem. Phys.*, **105,** 3604 (1996) for an RI approximation, and M. Challacombe and E. Schwegler, *J. Chem. Phys.*, **106,** 5526 (1997) for a multipole expansion approximation.
2. K. Ishida, *Int. J. Quantum Chem.*, **59,** 209 (1996).
3. K. Ishida, *J. Chem. Phys.*, **98,** 2176 (1993).
4. M. Head-Gordon and J. A. Pople, *J. Chem. Phys.*, **89,** 5777 (1988).

5. R. Lindh, U. Ryu, and B. Liu, *J. Chem. Phys.*, **95,** 5889 (1991).
6. T. P. Hamilton and H. F. Schaefer III, *Chem. Phys.*, **150,** 163 (1991).
7. L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.*, **26,** 218 (1978).
8. M. Dupuis, J. Rys, and H. F. King, *J. Chem. Phys.*, **65,** 111 (1976).
9. P. M. W. Gill and J. A. Pople, *Int. J. Quantum Chem.*, **40,** 753 (1991).
10. S. Ten-no, *Chem. Phys. Lett.*, **211,** 259 (1993).
11. P. M. W. Gill, M. Head-Gordon, and J. A. Pople, *Int. J. Quantum Chem. Symp.*, **23,** 269 (1989).
12. J. Almlöf and P. R. Taylor, *J. Chem. Phys.*, **86,** 4070 (1987).
13. J. Almlöf and P. R. Taylor, *J. Chem. Phys.*, **92,** 551 (1990).